

CPSC 1155 – Lab 10

C++ Functions

Lab Introduction

This lab helps you practice with functions in C++.

Learning Objectives

At the end of this lab, you should be able to write C++ functions.

Lab Readings

Chapter 6 – Functions

Lab Instructions

For each **Problem Statement**, follow the steps below:

1. Read the problem statement and clarify the problem.
 - a. Break the problem into smaller problems if needed.
2. Determine the IPO.
 - a. Determine input, output, intermediate variables, constants, conditions, and repetitions.
 - b. Declare the variables and constants (data type + meaningful names).
 - c. Work out the problem by hand using typical input values. Determine the range of valid input values.
 - d. Determine the process.
3. Write a function as required. For each function:
 - a. Determine the parameters and their data types.
 - b. Determine the data type for the returned value.
4. Write the test program (main) as required. Reading input values and displaying results happen in the test program.
 - a. Add comments where needed. Make sure to use a comments header to reflect the intention of your program and name of the author (you) and the date the program was written.
 - b. Test, debug, and execute the program using typical values.

Submit according to the instruction in the "Lab Submission" section.

Problem Statements

1. [5] (even_odd.cpp) You are required to determine if an integer is odd or even.

Write a **function**, called *evenOrOdd*, which receives an integer as parameter and returns True or False if the number is even or odd respectively. **Validation is not required.**

Write a test program that inputs an integer, uses *evenOrOdd* to determine if the integer is odd or even, and displays the results. Here are sample runs:

Enter an integer: 69

69 is odd

Enter an integer: 200

200 is even

2. [5] (temp.cpp) The National Weather Service implemented the wind-chill temperature to measure the coldness using temperature and wind speed. The formula is:

$$t_{wc} = 35.74 + 0.6215t_a - 35.75v^{0.16} + 0.4275t_av^{0.16}$$

where t_a is the outside temperature measured in degrees Fahrenheit and v is the speed measured in miles per hour and t_{wc} is the wind-chill temperature. The formula cannot be used for wind-speeds below 2 mph or temperatures below -58°F or above 41°F.

Write a **function**, called *windChill*, which receives the temperature in Fahrenheit and the wind speed as parameters, and returns the wind-chill temperature.

Write a test program that inputs a temperature and speed, tests their validity, calls *windChill* to calculate wind-chill temperature, and displays the results.

3. [5] (sum.cpp) Write a **function**, called *sum*, which receives two integers i1 and i2 as parameters, and calculates and returns the sum of integers from i1 to i2. For example, if i1 is 5 and i2 is 9, sum is 5+6+7+8+9.

Write a test program that reads two values for num1 and num2, uses *sum* to calculate the sum of numbers from num1 (the smaller number) to num2, and displays the results. **You need to make sure the first parameter is the smaller number.**

4. [5] (rand_char.cpp) You are required to generate a password consisting of random digits, uppercase and lowercase letters. The format is CCcdd where C stands for an uppercase letter, c for a lowercase letter, and d for a digit.

Write a **function**, called *randomCharacter*, which receives a start character and an end character as parameters, and returns a random character between the start and end characters inclusive. For example, *randomCharacter('C','M')* will generate and return a random character ranging from C to M.

Write a test program that uses *randomCharacter* to generate random characters and displays a random password with the above format. You need to call the function for every character.

5. [5] (encoding.cpp) You are required to encode a string by replacing some letters with digits as follows:

O or o with 0

L or l with 1

B or b with 8

Write a function, called *encodeString*, which receives a string as parameter, replaces the characters as explained above, and returns the new string. For example, *encodeString("I love Football")* will return "I 10ve f00t8a11".

Write a test program that reads a string, uses *encodeString* to encode the string, and displays the encoded string.

Lab Submissions

Submit a zip folder named as yourName_Lab10.zip to Brightspace. This folder should consist of five **C++ programs** for Problem Statements.

Please make sure that all your C++ programs compile and run properly before submission.

Marking Scheme

The marks are given in square brackets [] for each question.